Optimization Principles for Hardware/Software Co-Design with Applications in Molecular Dynamics

**Stephan J. Eidenbenz, CCS-3; Kei Davis, CCS-7;
Arthur F. Voter, T-1; Hristo N. Djidjev, Leonid Gurvits,
CCS-3; Christoph Junghans, T-1;
Susan M. Mniszewski, CCS-3; Danny Perez, T-1;
Nandakishore Santhi, Sunil Thulasidasan, CCS-3**

We are developing a framework for hardware-software co-design as a formally posed optimization problem. While the optimization framework will be applicable to multiple problem domains, our target application is molecular dynamics (MD), an exemplar for the need for both weak (larger problems) and strong (faster time to solution) computational scaling, and archetypal of the obstacles thereof. We view co-design as search and selection from a vast space of hardware and software designs that map to performance metrics such as run time (or computational rate), problem size, simulated time duration, energy use, and hardware cost. Successful implementation will provide the unique capability of co-designing hardware and software on a sound formal basis. It will enable effective implementation of physics simulations on future novel computer architectures, and provide a basis for more decisive influence on hardware design. For our MD application domain, we will define plausible regimes that would enable us to study a wealth of poorly understood phenomena of critical importance to the LANL mission.

Hardware-software co-design, perceived as a prerequisite to achieving effective exascale computing, needs to be put on a sound scientific basis such that design decisions for both hardware and software are not made based on colloquial heuristic insights, but rather follow an established scientific procedure by a sufficiently thorough search of realizable hardware and software options.

Figure 1 illustrates our approach, where the hardware (left) and software (right) design boxes represent the spaces of hardware and software solutions, whose combinations lead to performance predictions, the results of which in turn guide an optimization method towards new hardware/software solutions to be tested. The resulting iterative process can be analyzed in formal and informal settings, thus opening doors to established optimization and analysis techniques, while at the same time incorporating sometimes superior but informal human ingenuity. The key research efforts in this approach are:

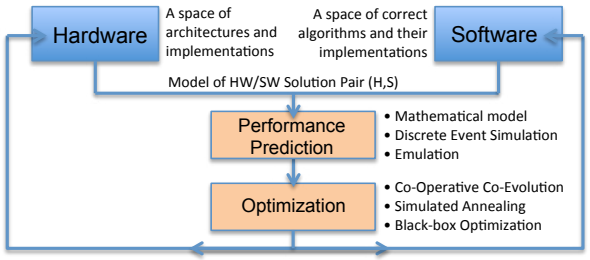

Fig. 1. The co-design optimization loop: the optimization method repeatedly applies the performance prediction method to the previously selected hardware/software pair, then selects a new pair based on that predicted performance.

- efficient enumeration strategies for both the hardware and software design spaces ("What are our options?") that accurately reflect physical constraints and trade-offs in realizable hardware and software designs;
- a multi-scale approach to performance prediction modeling ("How fast will it run?"), where we use cycle-accurate virtual machine emulation, discrete event simulation, graph mapping, and constraint programming as different prediction methodologies; and
- optimization methods to search the design spaces ("How do we select solutions?"), with fast identification of new HW/SW pairs to be tested with the more detailed performance prediction methods.

We define hardware and software designs in a hierarchical fashion. Enumeration of software designs will initially be via correctness-preserving transformations. The enumeration of hardware architectures is done implicitly through parameter spaces and optimization search methods as well as design feasibility checking (human, possibly machine-aided). Our performance prediction methods model at different levels of detail, thus spanning the trade-off between accuracy and scale in both time and size. The performance prediction techniques we will employ include constraint mathematical programming, discrete event simulation, and virtual machine emulation. Optimization heuristics are necessary for the more detailed prediction methods of virtual machine emulation and discrete event simulation. Our framework allows for the adaptation of standard meta-heuristics such as simulated annealing, taboo search, gradient search, and genetic algorithms. The more coarse-grained prediction methods of graph mapping and constraint

For more information contact Kei Davis at kei@lanl.gov

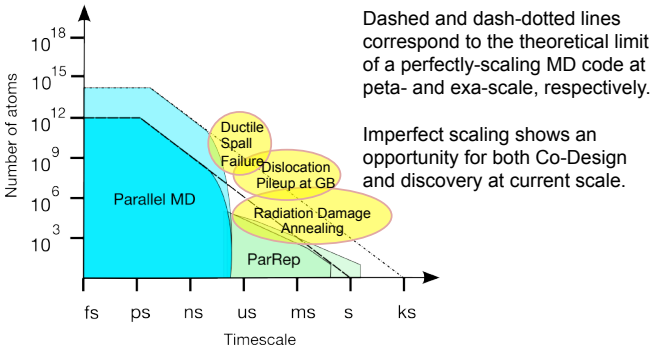

Fig. 2. The need for Co-design to overcome current performance boundaries of MD/AMD. Strongly colored regions correspond to the accessible problem space on a current petascale computer (24 hours on Roadrunner) while paler regions are for an exascale machine. The dashed and dash-dotted lines correspond to the theoretical limit of a perfectly scaling MD code at the peta- and exascale, respectively. Note the capability hole at 10^6–10^9 atoms for microseconds to milliseconds for both peta- and exascale.

programming will optimize through graph algorithms and mathematical programming techniques.

On the application side, our goal is to enable the atomistic simulation tools that can probe the physics of processes such as ductile spall failure under shock conditions, and the evolution of radiation damage. Achieving this requires two orders of magnitude increase in simulated time over current state-of-the-art petascale computing, but direct implementation on an exascale platform would not achieve it. These two material-science problems are of great intrinsic interest to LANL, addressing the response of materials in extreme conditions, and enabling the design of more durable and safe fission power plants.

The severity of the scaling problem becomes obvious when we estimate the behavior of MD or accelerated molecular dynamics (AMD) techniques [1-4] such as parallel-replica methods (ParRep [2]) on an anticipated realization (say in 2020) of an exascale machine with 10^8 processing cores (Fig. 2) [5]. The increased memory will allow simulations up to ~10^{14} atoms for picosecond time scales, and ParRep for very small systems might reach 1-second time scales if the events were very infrequent (once per 10 nanoseconds for 1000 atoms), but essentially no new territory is covered in the middle range. The important regime of 10^6 to 10^9 atoms for microseconds to milliseconds is still totally inaccessible, meaning we will have no direct way to simulate processes such as ductile spall failure, dislocation pileup and release at grain boundaries, or radiation damage annealing involving multiple cascades, all of which are important materials science problems and critical to the Department of Energy mission. In short, if we do not find a new paradigm for implementing MD methods on advanced computer architectures, critical problems in materials science that could in principle be addressed in a powerful way with MD or AMD will remain utterly out of reach for the indefinite future. We believe the answer to this problem lies in the development of a formal co-design capability that optimizes hardware and software simultaneously.

At the highest level this work will yield a sound basis for conducting disciplined hardware/software co-design as a formally-cast optimization problem. Within this conceptual framework we will provide a number of techniques, at various levels of abstraction, for modeling hardware and its performance. While these may not be fundamentally new, what will be new is their being posed such that they provide a well-defined and consistent interface to an optimizer. We will demonstrate the practical utility of using formal optimization techniques to rapidly identify high value points in the design space. While MD calculations are the focus, the framework and techniques are general.

A successful co-design optimization framework will fundamentally change the way scientific applications are constructed: hardware/software co-design fundamentals will become an essential part of software engineering. The optimization framework will have manual and automated aspects, similar to compiler optimization techniques used in classical compilers for the automated aspects; it will achieve peak performance when the human designer allows the optimizer to test a large set of potential solutions in a high-throughput optimization loop.

[1] Voter, A.F., *Phys Rev Lett* **78,** 3908 (1997).
[2] Voter, A.F., *Phys Rev B* **57,** R13985 (1998).
[3] Sorensen, M.R. and A.F. Voter, *J Chem Phys* **112,** 9599 (2000).
[4] Perez, D. et al., *Annu Rep Comput Chem* **5,** 79 (2009).
[5] Perez, D. et al., *ADTSC Science Highlights 2010,* 130, Los Alamos National Laboratory (2010).

Funding Acknowledgments
LANL Laboratory Directed Research and Development Program